- CONTROL STATEMENTS –

Statement (문)	Executable statement (실행문)	Sequential statement (순차문)		
		Control statement (제어문)	IF (조건문)	if, switch
			LOOP (반복문)	for, while,do~while
	Non-executable statement (비실행문)	Annotation (// /* */)		

1. IF STATEMENTS:

Description:	Example:

`if (condition) {`
`실행문;`
`} else if (condition2) {`
`실행문 2;`
`} else {`
`실행문 3;`
`}`
	`public static void main(String[] args) {`
	`int seoulLunchPrice = 4000;`
	
	`if(seoulLunchPrice>=7000) {`
	
	`        System.out.println("It's expensive");`
	
	`}else if(seoulLunchPrice>=6000){`
	`        System.out.println("Wish it was cheaper");`
	`}else if(seoulLunchPrice>=5000){`
	`        System.out.println("Perfect");`
	`}else {`
	`        System.out.println("It's too cheap");`

2. SWITCH STATEMENTS:

Switch statements select execution statement according to the value of the variable
- When programmer sets a variable, the computer compares the variable with the value of each case.
- When the two equalize, the program runs that case's command (executable statement).
- If there is no case with the same value, it runs the default command
 	
*The program runs until it meets a break, in which case, it escapes the whole switch command.
	`Int/String variable = 숫자/문자`
	
	`Switch (variable) {`
	`        case 값 1:`
	`        실행문 1;`
	`        break;`
	
	`        case 값 2:`
	`        실행문 2;`
	`        break;`
	
	`        default:`
	`        실행문 3;`
	`                        }`
		`public static void main(String[] args) {`
		
		`Scanner sc = new Scanner(System.in);`
		`        System.out.print("insert score: “);`
		
		`int hak = sc.nextInt();`
		`int hak = sc.nextInt();`
		`int temp = (hak==100)?hak-1:hak;`
		
		`switch(temp/10){`
		`        case 9:`
		`        System.out.println("A ");`
		`        break;`
		`        case 8:`
		`        System.out.println("B");`
		`        break;`
		`        case 7:`
		`        System.out.println("C ");`
		
		`        break;`
		`        case 6:`
		`        System.out.println("D ");`
		
		`        break;`
		`        case 5: case 4: case 3: case 2: case 1: case 0:`
		`        System.out.println("F ");`
		
		`        break;`

		default: System.out.println(“Not a valid number"); } }

3. FOR STATEMENTS:

- initial value: the variable is given a value (either int or string)
- condition: the condition is run and in the case it is satisfied for that value, it executes the command (실행문).
- Whether the condition is satisfied or not, the value of the variable is changed by the increase/decreased, indicated by the last
	For (initial value ; condition ; change) {
	실행문;
	
	//E.g – continue = skip that value but continue for the rest
	For (int i=9 ; i<10 ; i++) {
	if (i==5) {
	continue ; }}
	Output: 0 1 2 3 4 6 7 8 9
	
	//E.g – break = break out of the cycle
	For (int i=9 ; i<10 ; i++) {
	if (i==5) {
	break ;
	}
	}
	
	Output: 0 1 2

4. WHILE STATEMENTS:

The number of repetitions is decided by the condition and only when the condition is satisfied, the command (within the block) is executed**.**	**While (condition) {**
	실행문;
	}

4.1 DO ~ WHILE STATEMENTS:

The condition is compulsorily executed once and only if the condition is true, it repeats.	**Do {**
	실행문;
	) while (condition);

5. TERNARY OPERATOR:

The ternary operator consists of a condition that evaluates to either true or false, plus a value that is returned if the condition is true and another value that is returned if the condition is false.	**String/int variable = (condition)? "x": (condition 2)? "y": "z";**
	
	//x: what is puts in the variable if condition 1 is true
	//y: what is puts in the variable if condition 2 is true
	//z: what is puts in the variable if neither conditions are true.
	
	Example:
	
	public static void main(String [] args) {
	int age = 21;
	System.out.println("Age is: " + age);
	
	String msg = (age<0) ? "not a valid number": (age >=18) ? "adult" : "child";
	
	System.out.println(msg); }